

How-to Guide

Mobile Client Technology

How To... Use UI Library

Version 1.00 - August 2007

Applicable Release: CRM Mobile Technology 4.0 and 5.0

© Copyright 2007 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary. These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

1	Introduction.....	1
2	Usage of the UI Library.....	1
2.1	Sample Dialog	2

1 Introduction

The UI library is designed to provide a generic API to create and display runtime dialog windows with a number of additional features and services. These features include:

1. Mobile Client visual branding scheme applied to provide standard look-and-feel of controls and windows.
2. Built-in support for strings translation using the language server provided by the framework.
3. Internal handling of right-to-left environment setting and so on.

This library is available as a part of `UFDialogs.dll` for framework and plug-in developers.

2 Usage of the UI Library

This library represents a total runtime approach. This means that the creator of a dialog is not able to see a dialog during the creation process and no visual dialog modeling tools are available. The author of a dialog must write code to instantiate a new dialog window object, add necessary controls to it, and add specific behavior like processing user input and handling events.

The following steps should be performed in order to create and display a dialog:

1. Create a new dialog object by derivation:

For example,

```
public class dialogTest : dialogBase(modal, caption,
borderstyle, RTLSupport)
```

2. Add controls:

For example,

```
FTextBox tbUser = (FTextBox)mWindow.AddControl
(FControlTypes.fcTextBox, "name", "caption", location, size,
targetFrame);
tbUser.Caption = translate(nSpace, CAPTION_GUID_INFO);
```

3. Show the dialog:

For example,

```
base.ShowDialog()
```

4. Handling control events:

For example,

```
private void tbUser_Change ()
{
}
```

In the above mentioned code samples:

- **modal** is a Boolean parameter and means **modal** (true) or **modeless** (false) dialog.
- **nSpace** is a String parameter and determines a namespace which will be used by the dialog to retrieve language-dependent strings from the language server.
- **borderStyle** is an enumeration of type *FStyles* and means fixed-size or sizable dialog.

```
public enum FStyles
{
    fsFixed = 1,
    fsSizable = 2,
    fsNonCloseable = 4,
    fsCloseable = 8,
}
```

- **fcTextBox** specifies the control being created. It is an enumeration of values representing all supported controls.

```
public enum FControlTypes
{
    fcBrowser = 1,                (FBrowser)
    fcButton = 2,                 (FButton)
    fcTextBox = 3,                (FTextBox)
    fcCheckBox = 4,               (FCheckBox)
    fcComboBox = 5,              (FComboBox)
    fcImage = 6,                  (FImage)
    fcLabel = 7,                  (FLabel)
    fcListView = 8,               (FListView)
    fcRadioButton = 9,            (FRadioButton)
    fcProgress = 10,              (FProgressBar)
    fcFrame = 11,                 (FFrame)
    fcListBox = 12,               (FListBox)
    fcBlueBox = 13,              (FBlueBox)
    fcRadioButtonGroup = 14,      (FRadioButtonGroup)
}
```

The corresponding framework classes are mentioned in brackets. For the complete description of interfaces of all classes please refer to UI-related documentation.

- **targetFrame** - The controls of the dialog could be grouped together using *FFrame control*, that is several radio buttons can represent a single logical group. To group several controls the *FFrame control* should be created first and then be passed to *AddControl* function during the creation of controls to be grouped.

2.1 Sample Dialog

The following sample code shows how to create a dialog with two textboxes, one combo box and two buttons:

```
using System;
using System.Drawing;
using SAP.MT.FW.UI.UFDialogs.FControl;
using SAP.MT.FW.UI.EnumsPublic;
using SAP.MT.FW.UI.Common;

namespace SAP.MT.FW.UI.UFDialogs.Specialized
{
    public class dialogLogin : dialogBase,
        ExposedInterfaces.ILoginWindow
```

```

    {
#region Controls

    private FTextBox tbUser = null;
    private FTextBox tbPswd = null;
    private FComboBox tbLang = null;
    private FButton btnOK = null;
    private FButton btnCancel = null;
#end region

    private const int REGULAR_WIDTH = 275;

    public dialogLogin(): this(false)
    {}

    public dialogLogin(bool RTLSupport): base (true, "",
        EnumsPublic.FStyles.fsFixed |
        EnumsPublic.FStyles.fsCloseable, RTLSupport)
    {
        Size s;
        Point p;

        mWindow.ShowInTaskbar = true;

        mWindow.width = REGULAR_WIDTH;
        int width = (int)mWindow.width - LEFT * 4;

        p = new Point(LEFT, 10);
        s = new Size(width, HEIGHT);
        tbUser = (FTextBox) mWindow.AddControl
            (FControlTypes.fcTextBox, "tbUser", "", p, s);

        p = new Point(LEFT, tbUser.Location.Y +
            tbUser.size.Height + HTSPACE);
        s = new Size(width, HEIGHT);
        tbPswd = (FTextBox) mWindow.AddControl
            (FControlTypes.fcTextBox, "tbPswd", "", p, s);
        tbPswd.Password = true;
        tbPswd.charLimit = 8;

        p = new Point(LEFT, tbPswd.Location.Y +
            tbPswd.size.Height + HTSPACE);
        s = new Size(width, HEIGHT);
        tbLang = (FComboBox) mWindow.AddControl
            (FControlTypes.fcComboBox, "tbLang", "", p, s);
        tbLang.allowEdit = false;

        p = new Point(width/2 - BTNWIDTH - 25 + 10,
            tbLang.Location.Y + tbLang.size.Height + HTSPACE);
        s = new Size(BTNWIDTH, BTNHEIGHT);
        btnOK = (FButton)
            Window.AddControl(FControlTypes.fcButton, "btnOK",
                "", p, s);
        btnOK.Default = true;
    }
}

```

```

        p = new Point(width/2 + 25, btnOK.Location.Y);
        s = new Size(BTNWIDTH, BTNHEIGHT);
        btnCancel = (FButton) mWindow.AddControl
            (FControlTypes.fcButton, "btnCancel", "", p, s);
        btnCancel.Cancel = true;

        mWindow.height = btnOK.Location.Y + btnOK.size.Height
            + HTSPACE * 6;

        setUpCaptions();
        setUpEventHandlers();

        mWindow.FontSizeEnum =
            getAppSettingValueInt("FontSize");
    }

private void setUpEventHandlers()
{
    mWindow.activate +=
        new Dialog.activateHandler (mWindow_activate);

    btnOK.onClick += new NoArgEventHandler(btnOK_onClick);
    btnCancel.onClick +=
        new NoArgEventHandler(btnCancel_onClick);
}

private void setUpCaptions()
{
    mWindow.Caption = translate
        (nSpace, "UI_CONT_DLG_LOGIN_CAPTION");
    tbUser.Caption = translate
        (nSpace, "UI_CONT_DLG_LOGIN_USER_LBL_L16");
    tbPswd.Caption = translate
        (nSpace, "UI_CONT_DLG_LOGIN_PASSWORD_LBL_L16");

    tbLang.Caption = translate
        (nSpace, "UI_CONT_DLG_LOGIN_LANGUAGE_LBL_L16");
    btnOK.Caption = translate(nSpace, "UI_CONT_BTNS_OK_BTN_L5");
    btnCancel.Caption = translate
        (nSpace, "UI_CONT_BTNS_CANCEL_BTN_L10");
}

#region PublicAPIs
public override bool ShowDialog()
{
    return base.ShowDialog();
}

public void finalize ()
{
    mWindow.Unload();
    performCleanUp();
}

public string username

```

```

    {
        get {return tbUser.Text;}
        set {tbUser.Text = value;}
    }
#endregion

private void btnOK_onClick()
{
    mWindow.Hide();
}

private void btnCancel_onClick()
{
    mWindow.Hide();
}

private void mWindow_activate()
{
    if (tbUser.Text.Length == 0)
        tbUser.setFocus();
    else
        tbPswd.setFocus();
}

public override bool performCleanUp()
{
    if (mbCleanupPerformed) return true;

    if (!closed)
    {
mWindow.activate -=
        new Dialog.activateHandler(mWindow_activate);
btnOK.onClick -= new NoArgEventHandler(btnOK_onClick);
btnCancel.onClick -=
        new NoArgEventHandler(btnCancel_onClick);

        closed = true;
    }

    tbUser = null;
    tbPswd = null;
    tbLang = null;
    btnOK = null;
    btnCancel = null;

    return base.performCleanUp ();
}
}
}

```